FACULTY OF INFORMATION TECHNOLOGY AND ELECTRICAL ENGINEERING

Tuomas Varanka

A COMPARATIVE STUDY OF FACIAL MICRO-EXPRESSION RECOGNITION

Bachelor's Thesis
Degree Programme in Computer Science and Engineering
September 2019

Varanka T. (2019) A Comparative Study of Facial Micro-Expression Recognition.
University of Oulu, Degree Programme in Computer Science and Engineering, 34 p.

ABSTRACT

Facial micro-expressions are involuntary and rapid facial movements that reveal hidden emotions. Spotting and recognition of micro-expressions is a hard task even for humans due to their low magnitude and short duration compared to macro-expressions. In this thesis we look at why micro-expressions are important, datasets that contain micro-expressions for training of automatic systems, and how we can utilize modern computational methods to automatically recognize micro-expressions. Furthermore, we experiment with several representative methods in the literature and compare their performance.

Keywords: Affective computing, Facial expressions

Varanka T. (2019) Vertaileva tutkimus mikroilmeiden tunnistuksesta. Oulun yliopisto, Tietotekniikan tutkinto-ohjelma, 34 s.

TIIVISTELMÄ

Mikroilmeet ovat tahattomia ja nopeita kasvojen liikkeitä, jotka kertovat henkilön piilotetuista ilmeistä. Mikroilmeiden tunnistus ja luokittelu on vaikea tehtävä jopa ihmisille niiden lyhyen keston ja pienten liikkeiden takia verrattaessa makroilmeisiin. Tässä työssä tarkastelemme miksi mikroilmeet ovat tärkeitä, data-aineistoja, jotka sisältävät mikroilmeitä automaattisten systeemien opetukseen ja miten mikroilmeitä voidaan luokitella moderneilla laskennallisilla keinoilla. Lisäksi tarkastelemme ja testaamme eri keinoja kirjallisuudesta ja vertaamme niiden tuloksia.

Avainsanat: Affektiivinen laskenta, Ilmeet.

TABLE OF CONTENTS

FOREWORD

This work was conducted during the summer of 2019 at the Center for Machine Vision and Signal Analysis (CMVS), University of Oulu. I would like to thank both my supervisors Guoying Zhao and Jingang Shi for their help with both the writing procedure and programming, as well as the insightful discussions of the topic.

Oulu, September 23rd, 2019

Tuomas Varanka

LIST OF ABBREVIATIONS AND SYMBOLS

AU	Action Unit
CASME	Chinese Academy of Sciences Micro-Expressions
CNN	Convolutional Neural Network
DFT	Discrete Fourier Transform
ELBPTOP	Extended LBP-TOP
EVM	Eulerian Video Magnification
FDM	Facial Dynamics Map
FFT	Fast Fourier Transform
FPS	Frames Per Second
GLMM	Global Lagrangian Motion Magnification
HIGO	Histogram of Image Gradient Orientation
HOG	Histogram of Oriented Gradients
HOOF	Histogram of Optical Flow
HS	High Speed
LBP	Local Binary Pattern
LOSO	Leave-One-Subject-Out
MDMO	Main Directional Optical Flow
ME	Micro-Expression
MEVIEW	Micro-Expression VIdEos in the Wild
MOP	Mean Orthogonal planes
OF	Optical Flow
PCA	Principal Component Analysis
RGB	Red, Green, Blue
ROI	Region of Interest
SAMM	Spontaneous Actions and Micro-Movements
SC	Sparse Coding
SIP	Six Intersection Points
SMIC	Spontaneous Micro-Expression Corpus
STCLQP	Spatio-Temporal Completed Local Quantized Patterns
SVM	Support Vector Machine
TICS	Tensor Independent Color Space
TIM	Temporal Interpolation Model
TOP	Three Orthogonal Planes
C	SVM's penalty hyperparameter for the error term
$coef$	SVM's kernel's bias term hyperparameter
$degree$	SVM's kernel's degree hyperparameter
FN	False Negative
FP	False Positive
$F1$	$F1$-$score$
I	Intensity function
TP	True Positive
V_i	Movement vector to direction i
γ	SVM's kernel's weight term of the dot product

1. INTRODUCTION

Emotion is something that separates humans from computers. The ability to convey emotions shows for example whether a person likes or dislikes something. Emotions can be expressed to others using non-verbal clues like facial expressions, gestures, vocal expressions, biosignals, text, or even by the way we walk or act. A smile—or the activation of AU6 and AU12 (action unit[1]) as researches in the field of facial gesture analysis would say—might indicate that the person likes something or is feeling pleasant. This is not always the case though. In 1969, Ekman and Friesen [1] noticed a micro-expression (ME) of anger from a patient that was trying to convince her doctor into believing she was not ill by smiling—later it was found out that the patient was suicidal. Micro-expressions have the ability to show the truth about a person's feelings.

Artificial intelligence is often portrayed as a sort of a robot in the mass media. The similarities these robots often have is the ability to act human-like, *i.e.*, portray emotions. The current artificial intelligent systems fail to account the emotions of the user—for example a recommendation system could take in to account the emotions of the user in order to better recommend products. The use of automatic detection of micro-expressions extend beyond recommendation systems as micro expressions show the true emotions of a person. Applications include lie detection, clinical diagnosis, business negotiation, forensic investigation and security systems [2].

The thesis is structured as follows. This chapter provides motivation to as why one should care about micro expressions. In addition, basics of emotions, MEs and how they should be measured are discussed. Lastly this chapter will provide a high level view of an automatic ME analysis system. Chapter 2 provides information about typical datasets used in the research of MEs. A whole chapter is devoted to datasets as creation of them has been found cumbersome—one of the reasons is the need for multidisciplinary professionals as the captured emotion samples need to be coded correctly in the dataset. Chapter 3 gives a detailed description of how recognition, *i.e.*, classification of MEs into their corresponding classes, is accomplished with various methodologies. In Chapter 4 we provide experiments on some techniques discussed in Chapter 3. Chapter 5 concludes the thesis.

1.1. Facial Expressions and MEs

Emotions are often conveyed through facial expressions. The seven universal emotions are: *happy*, *sad*, *anger*, *fear*, *surprise*, *disgust* and *contempt* [3]. But how does one measure emotion or classify it to a given class? For example, when a person is smiling we can easily state that they are happy. But what if they smile a little less, are they still happy? At what point does one emotion change to another? This is a problem that occurs when discretizing values. Theories of continuous valued emotions have been developed [4], where the axis contain *arousal* and *valence*, *e.g.*, happiness typically achieves a high value in the valence axis but a lower value in the arousal when

[1] AUs are based on FACS (Facial action coding system) which classify different facial movements on a persons face. AU6 is a cheek raiser and AU12 is a lip corner puller. Happiness can be thought as the combination of these two.

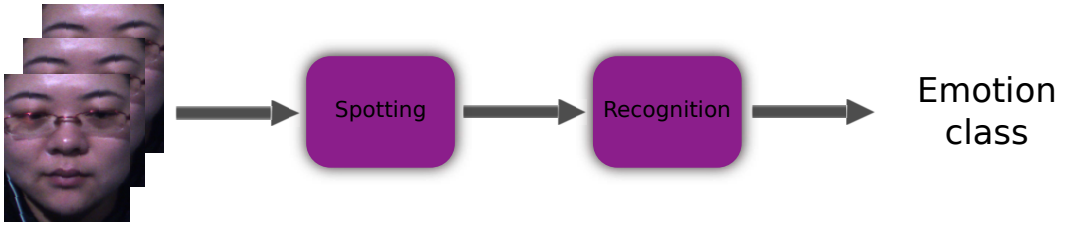

Figure 1. This figure showcases a general high level view of an ME analysis system. The system takes as input a sequence of videos (left). Spotting is performed to find the subset of frames where the ME is occurring. Recognition utilizes the information from spotting to classify the emotions to their corresponding classes. The number of emotion classes and what emotion classes are used depend on the dataset—these will be discussed in Chapter 2. Both of the spotting and recognition steps are typically complicated systems including preprocessing and many other steps. The example frames on the left are from the SMIC (defined later in Chapter 2) dataset and the images are shown with the permission of the authors of [8].

comparing to surprise. Often when analyzing emotions the discretized model is used, as it is easier to work with and potentially more natural.

MEs are short and involuntary facial expressions. In addition to being involuntary the person producing the ME might be unaware of its existence as well. It has been shown that MEs can not be posed, at least without practice, as posed MEs were found to be different in both the spatial and temporal domain in comparison to actual MEs [5]. Various intervals for the length of MEs have been proposed in the literature, but half a second seems to be the upper limit for an acceptable ME [2]. Another common characteristics of MEs are the low intensity and that only a part of the face is affected by the ME. The latter suggests implementations with ROIs (regions of interest) might succeed better in comparison to extracting features from the whole face at once.

1.2. Automatic ME Analysis System

To classify an ME from a sample one obviously needs to first be aware if there is an ME occurring at all, *i.e.*, spotting the ME. An automatic ME analysis system can be roughly divided into two sections: *firstly*, spotting an ME when it occurs; *secondly*, recognizing the emotion to its corresponding class[2] [7]. In real world applications the face might not be facing towards the camera, or simultaneous motion, such as blinking might happen at the same time as an ME. In addition to the resolution and/or the frame rate of the camera being low, creating an automatic ME analysis system is difficult as it comprises of many smaller tasks that all have to be working properly in order for the whole system to work. See Figure 1 for the workflow and steps of an ME analysis system.

[2]Recent work from [6] experiments with recognizing AUs, instead of the traditional emotion classes.

1.2.1. Spotting

Me spotting is the task of finding the subset of an video where the ME is occurring [2]. The spotted sequence should include the start of an ME (onset) and the end of the ME (offset). Alternatively, spotting can be done on the frame with the highest intensity of emotion (apex). Whether to use the onset and offset or the apex depends on the recognition method used.

An example of the onset and offset method is the feature difference method [9] that computes the difference in a feature space for all frames between the current frame and the average frame of tail and head frames in the specific sliding window. Having large movements in the current frame would yield a large dissimilarity measure between the current and the average frame. An example of apex spotting is [10], where the spotting is accomplished utilizing the frequency representation of the video. Transforming the video with 3D FFT (fast Fourier transform) and filtering the low frequencies (removing noise) reveals that the highest change of amplitude in the frequency corresponds to the apex frame. Frequency domain represents the change in pixels giving a much clearer look at changes in the video in comparison to the spatial domain.

1.2.2. Recognition

After spotting, the system knows the temporal interval where the MEs take place and the recognition part of the system can be used to classify the samples to one of the classes in the dataset. A popular type of recognition is to use optical flow (OF). Optical flow is a feature extraction technique that shows the movement in a sequence of videos. Aggregation of the OF values have been done in various ways [5, 11, 12, 13, 14, 15]. Another commonly used feature extraction techniques are the Local Binary Pattern (LBP) methods [7, 16, 17, 18]. After feature extraction, the samples can be classified. Most often used method has been the SVM (Support Vector Machine), with deep learning also becoming more often used recently [2]. A more detailed view of recognition methods are given in Chapter 3.

2. MICRO-EXPRESSION DATASETS

For any kind of machine learning approach a dataset is required to train the model. As ME analysis is still a relatively new field—not many datasets exist. In addition to being a new field, creating an ME dataset has been found challenging, which has also contributed to the amount of datasets. Due to low intensity and low temporal resolution, a high spatial resolution and a high frame rate is required for capturing videos. First attempts at creating datasets were from acted MEs. It has later been found out that acted MEs do not share the same characteristics as spontaneous ones [5] and using a model trained on acted MEs would most likely fail with real world spontaneous MEs. Thus, the focus has shifted away from acted datasets and towards spontaneous datasets.

Unfortunately, spontaneous datasets create their own problems which were not present with acted ones. Obviously, the subjects have to be emotionally involved. This is often achieved by showing carefully selected videos to the subjects that are meant to induce emotions—examples of the videos are: "Lion king" to induce *sadness* and "Funny cats" to induce *happiness* in the SMIC (Spontaneous Micro-expression Corpus) dataset. The subjects are told to suppress their emotions and a high stakes situation is created by giving penalties to people who show macro-expressions during the videos. In addition to inducing spontaneous MEs in data collection, labeling the videos is tedious, as labelling of onset, apex, and offset have to be done by looking at single frames. In addition, the labelling should be done by a professional who is qualified to distinguish between different MEs. These limitations also contribute to the low number of samples in the datasets. In this thesis we will focus only on the spontaneous datasets because of the reasons stated above. We will look at six different datasets, from which most of them can be characterized as being state of the art for ME datasets. A summary of the dataset can be seen from Table 1.

2.1. Spontaneous Micro-Expression Corpus (SMIC)

In [8], the authors extended the work of SMIC-sub [19], which consisted of 77 samples from 6 subjects captured at 100 FPS, to include three different versions with more samples:

(1) The high speed version includes 16 subjects and 164 samples captured at 100 FPS.

(2) The normal visual version was captured at only 25 FPS in attempts to try mimicking a normal security camera for example. It contains eight subjects with 71 samples.

(3) The near-infrared version was recorded side by side with the normal visual version and hence has the same number of samples and subjects.

An example from the dataset can be seen in Figure 2. All of the different versions were captured at a resolution of 640×480 and three classes were used to distinguish between different emotions: *positive*, *negative*, and *surprise* with the number of

Table 1. A summary of statistics from ME datasets

Dataset	Subset	Subjects	Samples	FPS	Resolution	Classes
SMIC [8]	HS	16	164	100	640 × 480	3
	VS	8	71	25	640 × 480	
	NIR	8	71	25	640 × 480	
CASME [20]	A	7	100	60	1280 × 720	8
	B	12	95	60	640 × 480	
CASME II [21]		26	247	200	640 × 480	5
CAS(ME)^2 [22]	B	22	57	30	640 × 480	4
SAMM [23]		32	159	200	2040 × 1088	7
MEVIEW [24]		16	31	25	1280 × 720	5

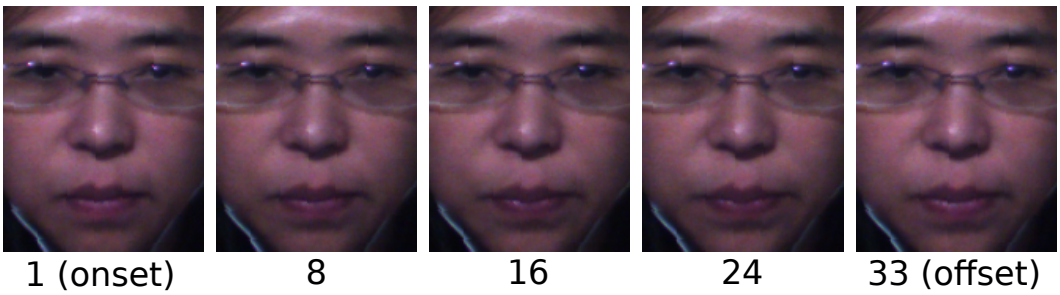

Figure 2. Example frames from the SMIC dataset from subject 1 showing surprise emotion. The numbers below correspond to the frame number. Images shown with the permission of the authors of [8].

samples for the high speed version: 51, 70, and 43, respectively. The dataset does not contain the annotation of AUs and the classes were reported by the subjects (self-report). The dataset originally contained 20 subjects but not all of them showed MEs and were therefore removed, leaving a total of 16 subjects.

2.2. Chinese Academy of Sciences Micro-Expression (CASME)

Constructed in [20], the CASME dataset contains 195 samples from 19 subjects. A lower frame rate of 60 FPS was used in comparison to the 100 in SMIC. Eight classes for emotions are used, seven of them being the basic emotions mentioned in Section 1.1 and the eighth being *tenseness*. A problem in CASME that was not in SMIC is the rather extreme imbalance of the class distribution. *Disgust* has 88 samples, while *fear* and *contempt* have only 2 and 3, respectively. See Table 2 for detailed distributions of emotion classes for all the datasets. The imbalance poses problems in both the used models for recognition and evaluation metrics. Solutions for the class imbalance problem will be discussed in Chapter 4.

Table 2. Distributions of emotion classes from ME datasets

SMIC [8]			
Subset	Positive	Negative	Surprise
HS	51	70	43
VIS	28	23	20
NIR	28	23	20

CASME [20] A & B							
Contempt	Disgust	Fear	Happiness	Repression	Tense	Sadness	Surprise
3	88	2	5	40	28	6	20

CASME II [21]				
Happiness	Disgust	Repression	Surprise	Others
33	60	27	25	102

CAS(ME)^2 [22]				
Subset	Positive	Negative	Surprise	Other
B	8	21	9	19

SAMM [23]							
Anger	Contempt	Fear	Disgust	Happiness	Other	Sadness	Surprise
57	12	8	9	26	26	6	15

MEVIEW [24]				
Anger	Contempt	Fear	Happiness	Surprise
1	6	6	8	9

2.3. CASME II

Improvements to the original CASME dataset were presented in CASME II [21]. The FPS was increased from 60 to a higher frame rate of 200. CASME had problems with the way the video clips were segmented as some of them were only 0.2 seconds long, having no frames before or after the ME, making spotting difficult. This was improved in the CASME II dataset. Flickering lights were also avoided, as these potentially create difficulties for feature extraction methods. The clips were also classified according to their corresponding AUs. A total of 247 samples were collected from 26 subjects. The number of classes were dropped to five due to the imbalance in the distribution of different emotions seen in CASME. The following emotions were kept: *happiness*, *disgust*, *surprise*, *repression*, and *others*. The distribution still remains somewhat unbalanced but not nearly as bad as in the case of CASME (see Table 2).

2.4. CAS(ME)^2

The authors of [22] created the CAS(ME)^2 dataset. It is a mixture dataset of both macro- and micro-expressions. Part A of the dataset contains both macro expressions and MEs. Part B contains 57 MEs from 22 subjects. The FPS was reduced to only 30 since both macro- and micro expressions needed to be captured. This dataset is not seen as commonly used as the others due to the low number of samples.

2.5. Spontaneous Actions and Micro-Movements (SAMM)

SAMM [23] consists of 159 samples from 32 subjects. It has 200 FPS as many of the other datasets but uses a very high spatial resolution of 2040×1088—giving better access to subtle movements. Another great thing about the SAMM dataset is its distribution of subjects. 13 different ethnicities are shown in comparison to the family of datasets from CASME, where all the subjects are from a single ethnicity, while SMIC has people from a total of three different ethnicities. The age distribution is also the highest with a mean of 33.24 (standard deviation: 11.32) years in SAMM, in comparison to 26.7 in SMIC (standard deviation not available, ages ranging from 22 to 34) and 22.03 (standard deviation: 1.60) in the CASME family of datasets. Low diversity in a dataset can create biases for the models. The samples are also labeled with the seven basic emotions as in CASME, but this time the labelling was done by a professional instead of self-evaluation by the subjects. The emotion inducing videos were selected personally to have higher chances of the subjects showing MEs. The personal videos were based on questionnaires. All in all, SAMM learned and improved from its predecessors on multiple fronts.

2.6. Micro-Expression VIdEos in the Wild (MEVIEW)

As opposed to the previously mentioned ME datasets which were all from laboratory settings, [24] collected a dataset from poker games and TV interviews on YouTube. The emotions were labeled by AUs and contained *contempt*, *surprise*, *fear*, *anger* and *happiness*. Since the videos are prerecorded and the environment could not be controlled, the video clips contain a variety of shots from different angles—making the preprocessing steps more difficult and more important.

2.7. Conclusion of Datasets

Typically the most used datasets in literature are CASME II and SMIC-HS. Both of the datasets contain high FPS, moderate spatial resolution, spontaneous MEs and a large amount of samples. Despite SAMM having arguably better features in comparison to CASME II and SMIC-HS, it is still not used as popularly. One of the reason for this might be the fact that it is rather new, as it was released in 2018 in comparison to SMIC in 2013 and CASME II in 2014. MEVIEW contains a more challenging dataset with different view angles and will probably become more popular as methods start achieving higher recognition rates on the laboratory controlled datasets, but the low number of samples would be a big limitation.

3. MICRO-EXPRESSION RECOGNITION

Recognition is the task of classifying the portrayed emotions to their corresponding classes, *e.g.*, a person is smiling, so most likely they are happy. The input to a ME recognition model is a video and the timestamps when the ME occurred—depending on the dataset this may include onset, offset and in some cases the apex. Hence, the MEs have to be spotted before the recognition phase can take place. Typically recognition consist of three main stages: preprocessing, feature extraction, and classification.

3.1. Preprocessing

Before feature extraction typically some kinds of preprocessing is performed. Most often this is face alignment after face detection, such that facial points at different time intervals can be found at the same locations through all the frames. Temporal domain interpolation is done in order to have all the samples the same size, as duration of MEs tends to vary. Due to the low intensity of MEs it makes sense to amplify the subtle movements—this is achieved with video motion magnification.

3.1.1. Face Detection and Registration

Ideally the face of subjects in the samples would stay still, but due to slight head movements different interest points may be located at different sections in different frames. Further, motion-based feature extraction methods will not work properly if an ME is happening at the same time as a head movement. The motion vectors will be overpowered by the head movements in comparison to the low intensity MEs.

To adjust a subject's face to a predefined position (typically the first frame of the video) specified landmarks are either manually or automatically detected and adjustments are made in the following frames. Different automatic landmark detection systems [2] include: Active Shape Model, Discriminative Response Maps fitting, Subspace Constrained Mean-Shifts, Face++, and Constraint Local Model [2]. Different techniques for face registration have been used: affine transformation, 2D-DFT and piecewise affine mapping, and locally weighted mean. Face alignment is typically done in the spatial domain, but for motion-based approaches the alignment can be accomplished in the motion domain for better results. In [5], 13 feature points are selected and an affine transformation matrix is calculated that minimizes the norm of the differences between the first frame's 13 feature points and the ith frame's points.

3.1.2. Temporal Domain Interpolation

Having different lengths of MEs can cause issues in recognition. If the samples are too short in duration they restrict feature extraction methods that utilize varied window lengths. On the other hand, if the samples are too long, redundant information may be packed due to the use of high frame rates. Also, classification methods typically require

the inputs to be of same size. To combat this [7] utilizes Temporal Interpolation Model (TIM) [25] to interpolate or to downsample depending whether the samples have less or more frames than the predefined length. They show that TIM increases performance, and the best results are achieved with 10 frames for SMIC. The average length of SMIC-HS is 33.7 frames, thus TIM significantly downsampled the video sequences. The results indicate that using all frames packs redundant information and not all of them are needed. Later work [10] show that only using the apex frame for recognition can also give promising results.

3.1.3. Video Motion Magnification

Motion magnification is an idea that makes intuitive sense—as a results of the low intensity of MEs, amplifying the motion would hopefully transfer the problem to what one could think of as macro-expression recognition. Unfortunately this is not necessarily the case since motion magnification also increases noise. Nevertheless, [7] shows that significant improvements can be accomplished. The main earlier method used in [7] was Eulerian Video Magnification (EVM) [26] which works by decomposing the image to multiple levels, after which each of the multiple levels are amplified by a magnification factor α. Global Lagrangian Motion Magnification (GLMM) [27] utilizes the whole sequences and gets a global approach as opposed to the EVM by utilizing a common reference point. A warping operator is also used to map the movements more precisely. Lastly, to avoid unnecessary noisy movements Principal Component Analysis (PCA) is used to remove small principal components.

3.2. Feature Extraction

Feature extraction is a key step in machine learning. Feature extraction is needed in image analysis to create distinguishable differences between different classes as typically images from their original domain do not contain discriminative features and often even contain redundant information. Traditional machine learning approaches rely on handcrafted features that are extracted from the input, while deep learning approaches learn the important features themselves from the data. Due to limited amount of datasets and samples many of the approaches used in ME analysis rely on handcrafted features as deep learning approaches typically require high number of samples. As with the rise of deep learning and recent advances in methods that do not rely on large amounts of data have made deep learning approaches more reasonable for ME analysis—some deep learning methods have even started to surpass the traditional machine learning methods. Many of the early feature extraction methods used were inspired from macro expression analysis, where they have achieved good results.

3.2.1. Local Binary Pattern -Based Methods

Local Binary Pattern (LBP) is a texture descriptor that thresholds neighboring pixels in a circular area from divided blocks—transforming the neighboring pixels to binary

code. Given that MEs occur both in the spatial and temporal domain LBP could not be used as it only considers the spatial domain. Hence, an extension of LBP called Local binary pattern on three orthogonal planes (LBP-TOP) [28] is used instead. LBP-TOP extends to the temporal domain by not only calculating the spatial plane (XY) but the vertical spatio-temporal plane (YT) and the horizontal spatio-temporal plane (XT) [2]. A concatenation of all the three different planes is done lastly. LBP-TOP has been one of the most popular feature extraction methods in ME analysis and serves as the typical baseline used when comparing new proposed methods. In addition, LBP-TOP also served as the baseline for many of the datasets described in Chapter 2.

Various different modifications and tweaks have been done to LBP-TOP to achieve improved scores. One of the simplest modifications is not to include all of the XY, XT, and YT planes, *e.g.*, LBP-XYOT utilizes the planes of XT and YT only—effectively reducing redundant information. LBP-SIP (Local binary pattern - six intersection points) [16] uses six points from the cross section of the three different domains that are unique to reduce redundant information—when capturing neighboring points in all the three domains some of the points are bound to be duplicated in LBP-TOP. LBP-MOP (Local binary pattern - mean orthogonal planes) reduces computational time by using means of the different domains rather than all the frames in the sample. LBP-MOP gives similar results to LBP-SIP but with the advantage of reduced computation [2]. In [17] the authors developed STCLQP (spatio-temporal completed local quantized patterns) which exploited more information from the sign, magnitude and orientation.

Extended LBPTOP (ELBPTOP) [18] is an extension to the original LBP-TOP feature extraction. ELBPTOP not only utilizes the first order information from LBP-TOP but takes advantage of the second order information from radial and angular differences. Radial difference uses a second ring that is lower in it's radius to the LBP and thresholds the difference between the corresponding elements from each of the rings. Angular difference thresholds the difference between the neighboring points in the first ring. Neither uses the center pixel for calculations. The radial and angular differences are then calculated in the three orthogonal planes and concatenated with LBP-TOP.

Other variants with different inputs, *i.e.*, the input image was modified through some transformation. An example of this is the TICS (Tensor independent color space) [29]. First, the samples were transformed to a tensor where the first two dimensions contain spatial information, the third temporal information, and the fourth color information. The color information is transferred from RGB to TICS where each of the components are as independent of each other as possible.

3.2.2. Optical Flow -Based Methods

Optical flow (OF) measures the difference between two sequences of images by detecting the intensity change of pixels. This can be mathematically formulated as follows according to the brightness constraint:

$$I(x, y, t) = I(x + \Delta x, y + \Delta y, t + \Delta t). \tag{1}$$

The intensity change of $I(x, y, t)$ at pixel (x, y) at time t can be be thought as adding the movement Δx and Δy in duration Δt. By assuming the movement is small, Equation (1) can be constructed with the Taylor series by linearizing the right side, from which we get the following equation:

$$\frac{\partial I}{\partial x}V_x + \frac{\partial I}{\partial y}V_y + \frac{\partial I}{\partial t} = 0, \tag{2}$$

where V_x is the movement in x direction and similarly V_y is the movement in y direction [5]. Equation (2) is problematic since we have two unknown variables and only one equation. Different methods for approximating optical flow have been developed. A simple method is to assume that the neighboring points share the same changes of V_x and V_y—this changes the problem to an overdetermined linear system which can be solved easily by least squares [30]. In practice more advanced methods are typically used with less assumptions. Examples of optical flow can be seen in Figure 3 and Figure 4. Figure 3 shows the change of magnitude of optical flow in the temporal domain—it can be seen that the highest magnitude is achieved close to the apex frame. Figure 4 shows how different emotion classes have different facial movements.

Optical flow in ME analysis

The temporal domain is essential in ME recognition, hence methods that directly measure the movement in sequences of images should make ideal candidates for feature extraction methods. One of the first approaches of using OF in ME analysis were by using a histogram of optical flow (HOOF) [13, 5]. HOOF is constructed by transforming the movement vectors $[V_x, V_y]$ from Cartesian coordinates to polar coordinates—giving the information of the angle and magnitude. The movement vectors are then binned to a histogram based on their orientation. Each of the bins magnitude is the sum of the movement vectors magnitude in that bin. For example when using 4 bins the overall movement of the scene can be observed into right, left, up, or down, based on the angle from polar coordinates.

Main directional mean optical flow (MDMO)

The authors of [5] improve HOOF by introducing MDMO (main directional mean optical flow). MDMO works by calculating the histograms on individual ROIs (regions of interest) based on AUs (action units)—these 36 ROIs were carefully constructed so that each of them would include at least one AU and they are by no means rectangular as used with many of the LBP methods. As opposed to the original HOOF, MDMO uses only the most important (main direction) feature vector. The most important feature vector is calculated as the mean of the feature vectors in the bin that has the highest number of vectors in it. This procedure eliminates redundant information by giving only a single movement direction for each of the ROIs.

To avoid motion vectors from being affected by slight head movements during the samples a face alignment procedure was applied. 13 feature points were chosen from the first frame and those points were then compared to the points in the ith frame. An affine transformation matrix was learned by looking at the difference of these

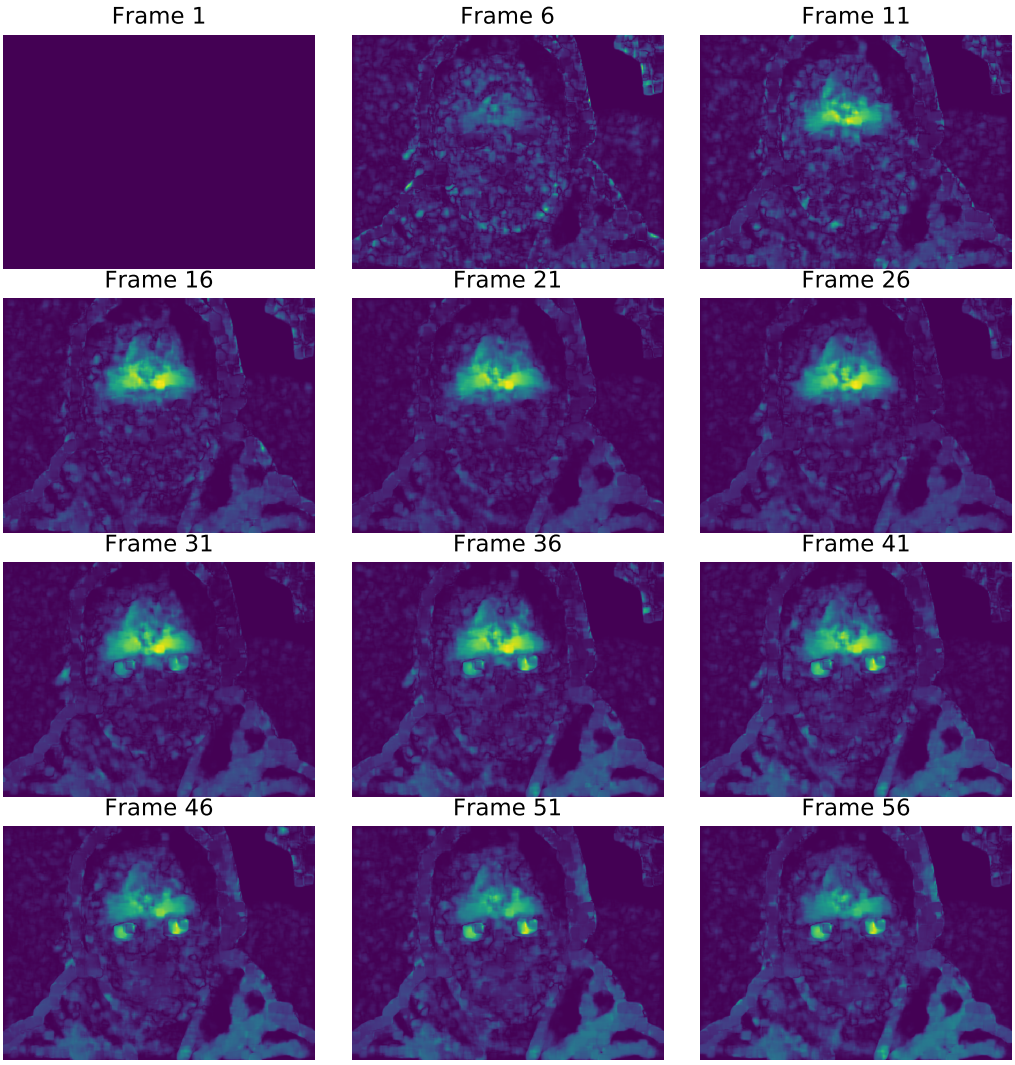

Figure 3. Frames of surprise ME shown in the optical flow domain (magnitude from the movement vectors). The sequence shows every fifth frame where the onset frame is the 1st frame, apex frame is the 30th frame and the offset frame is the 56th frame. The magnitude of the emotion can be seen as increasing until around the 30th frame after which the magnitude starts decreasing. The optical flow [31] is calculated between the first and the ith frame, hence the first frame is empty. Video from CASME II dataset [21] from subject 2 reacting to material "EP13_04".

points. Instead of the typical face alignment in the image domain, the procedure was instead done in the OF-domain—providing a more robust alignment and better results in typical cases. Light flickering can cause the movement vectors to show unwanted results as it measures the intensity. To avoid light flickering in SMIC and CASME a textural decomposition was used, where the original image was decomposed into two parts, structural part and the textural part. The textural part is unaffected by the light flickering and was thus used to compute the optical flow.

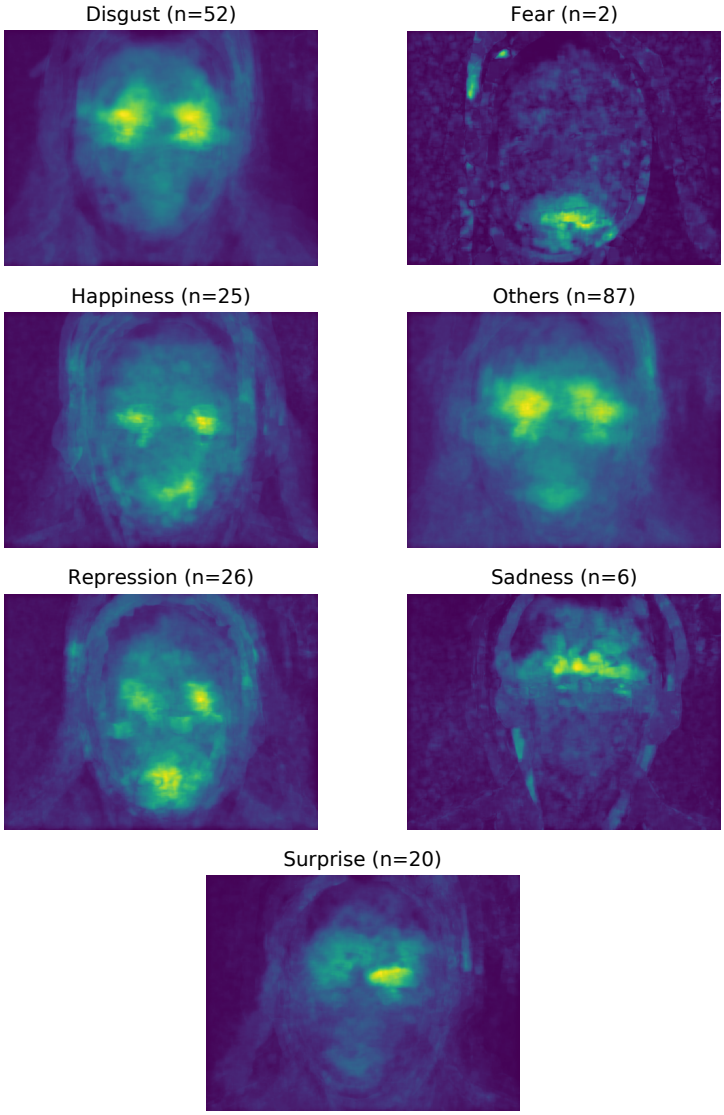

Figure 4. Indexing the samples based on their emotion class and calculating the average of the apex frames of optical flow to form a single image gives insights as to what facial muscles are active for each of the emotions. Disgust and surprise seem to mainly focus on eye movement, while *happiness*, *repression* and *others* have movements in both the eyes and mouth area. *Fear* and *sadness* both have low number of samples and definite conclusions are difficult to make. The shape of the head can be seen due to small movements and averaging a large number of frames together—this is less noticeable without averaging (see Figure 3). The n above the images stands for the number of samples averaged together. The samples are from the CASME II dataset [21].

Sparse MDMO

Sparse MDMO [12] continues the work of MDMO by introducing a sparse representation of the MDMO feature. The sparsity is obtained by utilizing GraphSC (Graph Sparse Coding) [32], an unsupervised dictionary learning algorithm. Averaging over the MDMO features of all frames in MDMO loses the underlying low dimensional

manifold structure. To reconstruct the manifold the authors propose a new distance metric for MEs that is able to distinguish different ME classes from each other. This distance metric is used as a regularizer in the sparsity coding—balancing between full reconstruction, preserving the low dimensional manifold, and sparsity.

Facial Dynamics Map

Facial dynamics map (FDM) [11] works similarly to MDMO in the sense that it tries to find principal directions from ROIs. The sample is first split into cuboids—rectangular in the spatial domain and the length of parameter τ in the temporal domain. FDM tries to maximize the sum of inner products in a given cuboid with the principal direction ξ. Since ξ is unknown, a starting value is given and the search is done iteratively. Face alignment is also performed in the OF-domain similarly to MDMO.

Bi-weighted oriented optical flow (Bi-WOOF)

In [14] the authors question the need for large amount of frames as they hold a considerable amount of redundant information and only utilize the onset and apex frame for recognition. A typical histogram approach of OFs is used as in MDMO and FDM with a slight twist. The samples (only the onset and apex) are divided into blocks similarly to FDM—there is no need for τ as only two frames are considered, thus the division to blocks is only done in the spatial domain. Unlike MDMO and FDM the optical strain is also utilized. The optical strain can be calculated from the optical flow map and it measures changes of length in a specific direction. In addition to weighting the histogram with the magnitude of the motion vectors, each block is weighted by the magnitude of optical strain. This procedure highlights histograms in blocks with large intensity of movement whereas noisy and small movements are left with low weights.

[15] continues the work of Bi-WOOF by adding phase information from Riesz transform from the frequency domain. The features from Bi-WOOF and phase are then concatenated together. Everything else is done essentially the same as in [14].

3.2.3. Deep Learning Based Methods

Deep learning based methods have seen an increase in the recent years in ME recognition [33]. Like many of the tasks in computer vision, ME recognition typically uses convolutional neural networks (CNNs). In addition to having an image aspect, ME recognition also contains a time-series, therefore typically requiring a sequence model. Combinations of LSTMs (long short-term memory) and CNNs as well as 3D CNNs have been used [3].

Due to the lack of large amounts of data, transfer learning seems a natural decision for using deep learning in ME analysis. Fine tuning a VGG-Face model was implemented in [10]. Unlike typical scenarios with large volumes of data the input images were not in their raw format, rather, Eulerian Video Magnification (EVM) described in Section 3.1.3 was utilized to enlarge the movements. The idea that the

apex frame contains the most important information for classification was used—also removing the need for a sequence model as the input only contains spatial information.

3.2.4. Other Methods

Several other feature extraction methods have also been tried in ME analysis that do not directly fall into LBP or OF -based methods. Histograms of oriented gradients (HOG) was used in [7], HOG calculates the gradient of the image, translates it in to polar coordinates and then quantizes the directions of gradients to bins. The bins are weighted by the magnitude of the gradient. A variant of HOG is HIGO (histograms of image gradient orientation) that does not weigh the bins. HIGO is a more robust method given illumination changes.

3.3. Classification

After preprocessing and feature extraction the actual recognition procedure can happen, *i.e.*, classification of emotions to their corresponding classes. By far the most used classifier in ME recognition has been the SVM (support vector machine) [2]. Other classification algorithms used include Adaboost, random forests, k-nearest neighbors, and extreme learning machines. SVM is known for its good classification performance and the fact that it works reasonably well in most cases has made it a safe choice in ME recognition.

4. EXPERIMENTS, ANALYSIS, AND DISCUSSION

This section contains experiments with selected methods from the literature that were previously discussed in Chapter 3. We provide experiments with the MDMO [5] and Sparse MDMO [12] as the representatives of the OF methods and [8] as the representative of LBP-TOP methods. The used evaluation metrics and techniques are discussed first.

4.1. Evaluation Methods

Proper evaluation techniques and metrics have to be used in order to compare methods with each other suitably. ME recognition is a classification task therefore typical evaluation techniques such as n-fold cross validation and accuracy can be used with minor tweaks.

4.1.1. Validation Techniques

Earlier works used the Leave-One-Video-Out (LOVO) validation [2] or more generally known as Leave-One-Out Validation (LOOV) or n-fold cross validation, where n is the number of samples in the dataset. LOVO takes a single sample and uses it as the testing set, while the rest are used as the training set. Evaluation is accomplished by training a model on the training set and evaluating on the testing set, and repeating the process for all samples. A problem with LOVO for ME recognition occurs from the personal bias of emotions, causing the model to possibly learn subject dependent features due to seeing the same person with possibly same emotions in both the training data and testing data. Therefore Leave-One-Subject-Out (LOSO) validation is often used to remove the personal bias by not having any samples in the training set from the subject in the testing set—yielding a more generalized metric [2]. LOSO works by selecting all the samples from a single subject as the testing set and the rest as the training set. A model is then trained from the training set and evaluated on the testing set. This procedure is performed for all the unique subjects in the dataset—also reducing the computational needs.

To better generalize ME recognition systems and have them more functional in situation outside the laboratory, [34] propose evaluating ME systems cross-database. Here, the training data is provided by a whole database and the testing data is provided by another whole database that is assumed to be a different database than the one used for the training data. Thus, the samples from the training data and testing data are drawn from different distributions—making the recognition task harder, but providing a model capable of generalizing better.

4.1.2. Metrics

Due to the imbalanced distribution of emotions in the datasets (see Table 2), the typical metric of *accuracy* is not used as it can be overwhelmed in a highly imbalanced situation. Often used metrics in an imbalanced situation are $Precision$ and $Recall$:

$$Precision = \frac{TP}{TP + FP}, \tag{3}$$

$$Recall = \frac{TP}{TP + FN}. \tag{4}$$

In Equations (3) and (4) TP refers to true positives, FP to false positives and FN to false negatives. In practice comparing two different metrics can be cumbersome, therefore a combination of the two is often used:

$$F1\text{-}score = \frac{2TP}{2TP + FP + FN}, \tag{5}$$

which can be considered as the harmonic mean between $Precision$ and $Recall$. As the classification is a multi-class problem and heavily imbalanced in some datasets—the $F1\text{-}score$ is macro-averaged. The macro-average $F1\text{-}score$ can be calculated by first calculating all the individual $F1\text{-}scores$ for each class and then taking the average:

$$F1\text{-}score_{macro} = \frac{1}{N}(F1_{class1} + F1_{class2} + \cdots + F1_{classN}),$$

where N refers to the number of classes. We simplify the notation and simply refer to $F1\text{-}score_{macro}$ as the $F1\text{-}score$ or $F1$ for short, in further analysis.

In ME analysis both accuracy and $F1\text{-}score$ (Equation (5)) are typically used side by side.

4.2. Implementation

The implementation for each of the methods are based on the code from the publications. We modify each of the codes to include the datasets of SMIC, CASME, CASME II and SAMM. Due to computationally heavy operations some of the hyperparameters are set fixed based on the original publications.

4.2.1. LBP-TOP

LBP-TOP [8] serves as a baseline method as it was the first method evaluated on SMIC. The method is simple: as input the algorithm takes cropped images, which is followed by interpolation by TIM and lastly the features are extracted by LBP-TOP. The code for

LBTP-TOP was more robust in comparison to the others, as the only required input was the dataset itself. This allowed us to utilize the original Matlab code heavily, with only minor tweaks. We add the functionality to load different datasets, tweak the LBP-TOP slightly and add the proper evaluation methods. We set the non-SVM hyperparameters as used in the original publication [8] for all the datasets: TIM normalized frames was set to 10 and the LBP-TOP divided blocks was set to $(8 \times 8 \times 1)$, corresponding to rows, columns and temporal blocks, respectively. We hypothesize that using a longer normalized frame length may improve the result for other datasets that use a higher frame rate.

4.2.2. MDMO

The original code for MDMO was developed in Matlab and it had some restrictions as facial feature points were used as an input, only two datasets were used; where not all the participants were included and the lack of evaluation methods. The code was fully replicated in Python, with slight modifications as the exactly same method for calculating optical flow could not be found in Python. We set the non-SVM hyperparameter λ (see [5] for details), which is a weight between the convex combination of the magnitude and angle defined in Section 3.2.2, to 0.9 for all the datasets. The normalized frame length was set to none, meaning that the frames were not normalized at all.

4.2.3. Sparse MDMO

Again, the original code for Sparse MDMO was developed in Matlab and it had further restrictions. The input used was pre-calculated MDMO features, which were only included for three datasets, in addition to removal of some subjects (due to facial landmark detector not being able to correctly detect all landmarks, the same as MDMO). In our implementation we utilize both the previously created Python version of MDMO, which allows to use all the subjects and more datasets, and the original code for Sparse MDMO, where the sparse mapping and classification is done. The sparse learning requires the input to be same size, thus an interpolation method is required. We extend our Python version of MDMO by adding TIM to normalize the frames. To summarize: *firstly*, the frames are normalized using TIM in Python; *secondly*, the MDMO features are calculated in Python; *lastly*, the calculated MDMO features are transferred to Matlab, where the sparse coding and classification is performed. We set the non-SVM hyperparameters the same as in [12]. As SAMM was not included in the original publication, the hyperparameters are set to the same as CASME II, as they share the same characteristics. For details of the hyperparameter settings refer to the original publication [12].

4.2.4. Hyperparameter Optimization

In the classification phase, we optimize the hyperparameters for SVM as the results varied highly on different hyperparameters. Grid search is utilized as it is easy to implement and the chosen search space is manageable with the following sets: $degree \in [2, 5]$ with an interval of 1, $coef \in [0, 3]$ with an interval of 1, $C \in \{0.1, 1, 10, 100\}$ with a logarithmic scale and $\gamma \in (0, 1]$ with an interval of 0.01. The coefficients $degree$, $coef$ and γ interact with each other in the following way:

$$K(x) = (\gamma x^T x + coef)^{degree},$$

where K is the kernel function used in SVM [35]. C is the penalty parameter for error terms. The sets for potential hyperparameters are based on previous work of [5, 12]. For LBP-TOP and SparseMDMO we restrict the search of γ's interval to 0.1 due to longer training times caused by the larger feature sets in comparison to MDMO. The hyperparameter optimization was done in terms of the best $F1\text{-}score$. We refer to the optimized hyperparameters with a quadruple $(degree, coef, \gamma, C)$ in the sections below.

All of the tested methods have many hyperparameters (*i.e.*, MDMO's λ, normalized frame length n_{frames} for all, LBP-TOP's hyperparameters and GraphSC's hyperparameters for SparseMDMO) that were not optimized due to long run times and were based on the previous works. We hypothesize that not correctly optimizing these hyperparameters can significantly impact the results.

4.3. Results

Table 3 summarizes the results from the experiments. We define a baseline with a dummy classifier—a dummy classifier classifies all the samples to the class with highest number of samples. Note the results for dummy classifier: quite a high accuracy caused by the imbalance of datasets, but also the low score on $F1$ due to high number of classes. This also emphasizes the use of $F1\text{-}score_{macro}$ instead of $F1\text{-}score_{weighted\_macro}$, that weights the classes based on the number of samples in them. The results would have been for weighted macro-averaging $F1\text{-}score_{weighted\_macro} = [0.26, 0.20, 0.23, 0.19]$—not indicating properly the difficulty of the problem's multi-class and imbalanced settings (compare to Table 3 Baseline $F1$).

The datasets used vary slightly from the discussion of Chapter 2—the possible differences are mentioned in the corresponding subsections below.

4.3.1. SMIC

The SMIC dataset is used as it was presented in Chapter 2. SMIC is possibly the easiest dataset due to the low number of classes (3) and relatively balanced distributions (see Table 2). The baseline method also indicates this: 0.4268 accuracy and 0.1994 $F1$. The best result is achieved by Sparse MDMO on both accuracy and $F1$, with

Table 3. A summary of all methods on all datasets

Accuracy				
Method	SMIC	CASME	CASME II	SAMM
Baseline	0.4268	0.3651	0.4008	0.3585
LBP-TOP	0.4756	0.3968	0.3644	0.3333
MDMO	0.5366	0.4233	0.5223	0.4654
Sparse MDMO	0.6098	0.4709	0.4939	0.4528
$F1\text{-}score$				
	SMIC	CASME	CASME II	SAMM
Baseline	0.1994	0.0669	0.1145	0.0660
LBP-TOP	0.4173	0.2573	0.2565	0.1989
MDMO	0.4736	0.2799	0.4699	0.2837
Sparse MDMO	0.5547	0.3258	0.4481	0.3261

the results 0.609 and 0.5547, respectively. These results were achieved with the hyperparameters $(5, 1, 0.41, 1)$. Closely after is MDMO and lastly the LBP-TOP as one would expect based on the results from the original publications [12, 5, 8]. For LBP-TOP and MDMO the optimized hyperparameters are the following $(2, 1, 0.01, 0.1)$ and $(2, 1, 0.39, 0.1)$, respectively.

4.3.2. CASME

From CASME only 189 samples are used instead of all the 195. Despite having all of the 195 samples, only 189 of them included the metadata—forcing us to only use the 189 samples. With 8 classes and highly imbalanced distribution, the baseline indicates this well: achieving a 0.3651 accuracy and a miserable $F1$ of 0.0669. Similar results occur in CASME as they did in SMIC in terms of the ranking of the methods. Sparse MDMO achieves the best result with an accuracy of 0.4709 and $F1$ of 0.3258, with the hyperparameters $(3, 3, 0.21, 1)$. MDMO falls behind Sparse MDMO by a margin of 0.0459 in terms of $F1$ in absolute difference, with the optimized hyperparameters $(5, 1, 0.7, 0.1)$. Lastly is LBP-TOP with the optimized hyperparameters being $(5, 1, 0.01, 0.1)$.

4.3.3. CASME II

CASME II actually includes a total of 256 samples, but typically 9 of them are cut off as these belong sadness and fear, which have a low sample number. We are left with 247 samples with 5 classes as mentioned in Chapter 2. With a 0.4008 accuracy and a 0.1145 $F1$ as the baseline, the same trend continues—a decent accuracy due to imbalance but a low $F1$ due to high number of classes. The best result is achieved by MDMO by a small margin to Sparse MDMO with a score of 0.5223 in accuracy and 0.4699 in $F1$ using the hyperparameters $(5, 2, 0.43, 0.1)$. This result is unexpected for Sparse MDMO and may be due to poorly optimized non-SVM hyperparameters or a simple bug in the program. The result of Sparse MDMO falls behind by only a small margin of 0.0218 $F1$ in absolute value using the hyperparameters $(3, 2, 0.41, 1)$.

It may seem that LBP-TOP fails in the task by achieving a lower accuracy than the baseline method. However, the $F1$ is significantly better than that of baseline's. This is due to optimizing the hyperparameters in terms of $F1$. The result for LBP-TOP was achieved with the hyperparameters $(5, 1, 0.01, 0.1)$.

4.3.4. SAMM

In SAMM we use 159 samples, the same as in Chapter 2, but instead use 8 classes as this is what our used metadata had. The added class is "Others". The baseline is very similar to CASME, with a 0.3585 accuracy and only a 0.0660 $F1$. The results in SAMM seem more familiar with SMIC and CASME in terms of the ranking of the methods. In terms of numerical results, SAMM shares similarities with CASME due to having the same number of classes. Sparse MDMO is able to attain an accuracy of 0.4528 which is slightly behind the value of MDMO's accuracy of 0.4654. However, Sparse MDMO achieves a better $F1$ score by a somewhat significant margin with the $F1$ being 0.3261, while that of MDMO's $F1$ is only at 0.2837. Since the hyperparameter optimization was done in terms of $F1$ we conclude that Sparse MDMO is superior in SAMM. We hypothesize that if the optimization was done in terms of accuracy, Sparse MDMO would still achieve a superior performance. The hyperparameters used for Sparse MDMO and MDMO were $(5, 3, 0.11, 0.1)$ and $(2, 2, 0.69, 0.1)$, respectively. LBP-TOP attains a similar result in SAMM as it did in CASME II—falling behind the baseline in terms of accuracy, but achieving a superior score in $F1$. The hyperparameters used for LBP-TOP were $(5, 1, 0.01, 0.1)$.

4.4. Comparison

Out of all the methods Sparse MDMO performs the best over all on all of the datasets, with the exception being MDMO on CASME II, where MDMO beats Sparse MDMO in $F1$ by a small, but nonetheless a significant margin. Datasets with higher number of classes (8): CASME and SAMM only achieve $F1$ of 0.3258 and 0.3261, respectively at their best. While, datasets with lower number of classes (3, 5): SMIC and CASME II achieve results of 0.5547 and 0.4699 $F1$, respectively at their best. It can be seen that the number of classes has a clear impact in the result. Overall, as can be seen from the results the task of ME recognition is difficult, as the best result achieves only a 0.6098 accuracy and a $F1$ of 0.5547.

The simple LBP-TOP can also be seen as being insufficient in terms of creating distinguishable features as most of the hyperparameter combinations tested resulted in the same values for accuracy and $F1$, this can also be seen from the sets of optimized hyperparameters for LBP-TOP mentioned above. The "optimized" hyperparameter quadruples mentioned for LBP-TOP were actually just one of the combinations that optimized the result.

4.5. Challenges and Discussion

As ME analysis is still in it's infancy both the results and evaluation methods are still lacking. Evaluation methods across different papers are extremely inconsistent. Different metrics, evaluation techniques and input are used—making comparison of results difficult. Reproduction of results are difficult due to not having the code publicly available. Inconsistencies in the datasets and their corresponding metadata, and self-evaluation based coding of emotions makes us wonder if the labels are correct. Work from [36] tries solving this problem by labeling the samples based on action units, making the task significantly easier, but possibly leading to incorrect results as different MEs may share the same set of action units [37]. A major problem with the datasets is also their size, not having enough training data can significantly decrease the performance of used methods. Furthermore, imbalanced nature of the datasets creates problems with the methods used for recognition, but also the use of metrics as can be seen from above. Recent research seems to be heading away from hand-crafted features and towards deep learning based methods [33] due to the rise of successful applications of deep learning.

Ethics of a system capable of spotting, recognizing and analyzing a person's emotions are to be thought carefully. MEs are involuntary and occur due to the person trying to hide their real emotion—this is highly private information that many would most likely not like to share. Use of lie detection systems have been deemed acceptable in high-stakes situations, but the person is aware of the situation in comparison to an ME analysis system that might recognize one's emotions from a security camera. Ethical issues regarding such systems are rarely discussed in the literature of automatic ME recognition, but should be taken into consideration as well.

5. CONCLUSION

This thesis covers the basics needed for micro-expression recognition. In the introduction we present motivation for an ME analysis systems, introduce basic concepts of emotions and MEs. We discuss a variety of methods ranging from appearance based hand-crafted features to deep learning. An exhaustive analysis of the ME datasets is conducted with insights as to what evaluation methods and metrics should be used. Finally, we present experiments with LBP-TOP, MDMO and SparseMDMO, and compare and discuss the results.

6. REFERENCES

[1] Ekman P. & Friesen W.V. (1969) Nonverbal leakage and clues to deception. Psychiatry 32, pp. 88–106. URL: https://doi.org/10.1080/00332747.1969.11023575, pMID: 27785970.

[2] Oh Y., See J., Ngo A.C.L., Phan R.C. & Baskaran V.M. (2018) A survey of automatic facial micro-expression analysis: Databases, methods and challenges. CoRR abs/1806.05781. URL: http://arxiv.org/abs/1806.05781.

[3] Merghani W., Davison A.K. & Yap M.H. (2018) A review on facial micro-expressions analysis: Datasets, features and metrics. CoRR abs/1805.02397. URL: http://arxiv.org/abs/1805.02397.

[4] Kollias D. & Zafeiriou S. (2018) Aff-wild2: Extending the aff-wild database for affect recognition. CoRR abs/1811.07770. URL: http://arxiv.org/abs/1811.07770.

[5] Liu Y., Zhang J., Yan W., Wang S., Zhao G. & Fu X. (2016) A main directional mean optical flow feature for spontaneous micro-expression recognition. IEEE Transactions on Affective Computing 7, pp. 299–310.

[6] Li Y., Huang X. & Zhao G. (2019) Micro-expression Action Unit Detection withSpatio-temporal Adaptive Pooling. arXiv e-prints , arXiv:1907.05023.

[7] Li X., Hong X., Moilanen A., Huang X., Pfister T., Zhao G. & Pietikäinen M. (2015) Reading hidden emotions: Spontaneous micro-expression spotting and recognition. CoRR abs/1511.00423. URL: http://arxiv.org/abs/1511.00423.

[8] Li X., Pfister T., Huang X., Zhao G. & Pietikäinen M. (2013) A spontaneous micro-expression database: Inducement, collection and baseline. In: 2013 10th IEEE International Conference and Workshops on Automatic Face and Gesture Recognition (FG), pp. 1–6.

[9] Moilanen A., Zhao G. & Pietikäinen M. (2014) Spotting rapid facial movements from videos using appearance-based feature difference analysis. In: 2014 22nd International Conference on Pattern Recognition, pp. 1722–1727.

[10] Li Y., Huang X. & Zhao G. (2018) Can micro-expression be recognized based on single apex frame? In: 2018 25th IEEE International Conference on Image Processing (ICIP), pp. 3094–3098.

[11] Xu F., Zhang J. & Wang J.Z. (2017) Microexpression identification and categorization using a facial dynamics map. IEEE Transactions on Affective Computing 8, pp. 254–267.

[12] Liu Y., Li B. & Lai Y. (2018) Sparse mdmo: Learning a discriminative feature for spontaneous micro-expression recognition. IEEE Transactions on Affective Computing , pp. 1–1.

[13] Chaudhry R., Ravichandran A., Hager G. & Vidal R. (2009) Histograms of oriented optical flow and binet-cauchy kernels on nonlinear dynamical systems for the recognition of human actions. In: 2009 IEEE Conference on Computer Vision and Pattern Recognition, pp. 1932–1939.

[14] Liong S.T., See J., Wong K. & Phan R.C.W. (2018) Less is more: Micro-expression recognition from video using apex frame. Signal Processing: Image Communication 62, pp. 82 – 92. URL: http://www.sciencedirect.com/science/article/pii/S0923596517302436.

[15] Liong S. & Wong K. (2017) Micro-expression recognition using apex frame with phase information. In: 2017 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC), pp. 534–537.

[16] Wang Y., See J., Phan R.C.W. & Oh Y.H. (2015) Lbp with six intersection points: Reducing redundant information in lbp-top for micro-expression recognition. In: D. Cremers, I. Reid, H. Saito & M.H. Yang (eds.) Computer Vision – ACCV 2014, Springer International Publishing, Cham.

[17] Huang X., Zhao G., Hong X., Zheng W. & Pietikäinen M. (2016) Spontaneous facial micro-expression analysis using spatiotemporal completed local quantized patterns. Neurocomputing 175, pp. 564 – 578. URL: http://www.sciencedirect.com/science/article/pii/S0925231215015726.

[18] Guo C., Liang J., Zhan G., Liu Z., Pietikäinen M. & Liu L. (2019) Extended Local Binary Patterns for Efficient and Robust Spontaneous Facial Micro-Expression Recognition. arXiv e-prints , arXiv:1907.09160.

[19] Pfister T., Xiaobai Li, Zhao G. & Pietikäinen M. (2011) Recognising spontaneous facial micro-expressions. In: 2011 International Conference on Computer Vision, pp. 1449–1456.

[20] Wen-Jing Yan, Wu Q., Yong-Jin Liu, Su-Jing Wang & Fu X. (2013) Casme database: A dataset of spontaneous micro-expressions collected from neutralized faces. In: 2013 10th IEEE International Conference and Workshops on Automatic Face and Gesture Recognition (FG), pp. 1–7.

[21] Yan W.J., Li X., Wang S.J., Zhao G., Liu Y.J., Chen Y.H. & Fu X. (2014) Casme ii: An improved spontaneous micro-expression database and the baseline evaluation. PLOS ONE 9, pp. 1–8. URL: https://doi.org/10.1371/journal.pone.0086041.

[22] Qu F., Wang S., Yan W., Li H., Wu S. & Fu X. (2018) Cas(me)^2: A database for spontaneous macro-expression and micro-expression spotting and recognition. IEEE Transactions on Affective Computing 9, pp. 424–436.

[23] Davison A.K., Lansley C., Costen N., Tan K. & Yap M.H. (2018) Samm: A spontaneous micro-facial movement dataset. IEEE Transactions on Affective Computing 9, pp. 116–129.

[24] Petr Husak Jan Cech J.M. (2017) Spotting facial micro-expressions “in the wild”. In: 22nd Computer Vision Winter Workshop.

[25] Zhou Z., Zhao G. & Pietikäinen M. (2011) Towards a practical lipreading system. In: CVPR 2011, pp. 137–144.

[26] Wu H.Y., Rubinstein M., Shih E., Guttag J., Durand F. & Freeman W.T. (2012) Eulerian video magnification for revealing subtle changes in the world. ACM Transactions on Graphics (Proc. SIGGRAPH 2012) 31.

[27] Le Ngo A.C., Johnston A., Phan R.C.. & See J. (2018) Micro-expression motion magnification: Global lagrangian vs. local eulerian approaches. In: 2018 13th IEEE International Conference on Automatic Face Gesture Recognition (FG 2018), pp. 650–656.

[28] Zhao G. & Pietikainen M. (2007) Dynamic texture recognition using local binary patterns with an application to facial expressions. IEEE Transactions on Pattern Analysis and Machine Intelligence 29, pp. 915–928.

[29] Wang S., Yan W., Li X., Zhao G. & Fu X. (2014) Micro-expression recognition using dynamic textures on tensor independent color space. In: 2014 22nd International Conference on Pattern Recognition, pp. 4678–4683.

[30] Sun D., Roth S. & Black M.J. (2014) A quantitative analysis of current practices in optical flow estimation and the principles behind them. International Journal of Computer Vision 106, pp. 115–137. URL: https://doi.org/10.1007/s11263-013-0644-x.

[31] OpenCV documentation for optical flow using Farneback’s method. URL: https://docs.opencv.org/2.4/modules/video/doc/motion_analysis_and_object_tracking.html#calcopticalflowfarneback. Accessed 24.7.2019.

[32] Nayak K., Wang X.S., Ioannidis S., Weinsberg U., Taft N. & Shi E. (2015) Graphsc: Parallel secure computation made easy. In: 2015 IEEE Symposium on Security and Privacy, pp. 377–394.

[33] Arxiv search for micro-expressions. URL: https://arxiv.org/search/?query=%22micro+expression%22+OR+%22micro-expression%22+OR+%22micro+expressions%22&searchtype=all&abstracts=show&order=-announced_date_first&size=50. Accessed 22.7.2019.

[34] Zong Y., Huang X., Zheng W., Cui Z. & Zhao G. (2017) Learning a target sample re-generator for cross-database micro-expression recognition. CoRR abs/1707.08645. URL: http://arxiv.org/abs/1707.08645.

[35] Sklearn documentation for SVM. URL: https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html. Accessed 20.8.2019.

[36] Davison A.K., Merghani W. & Yap M.H. (2017) Objective classes for micro-facial expression recognition. CoRR abs/1708.07549. URL: http://arxiv.org/abs/1708.07549.

[37] Lim C.H. & Goh K.M. (2017) Fuzzy qualitative approach for micro-expression recognition. In: 2017 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC), pp. 1669–1674.